CLAIMS

1. A computer readable medium having stored thereon a data structure that describes what types of binaries can be loaded into a process space for a trusted application, the data structure comprising:

a first portion including data representing a unique identifier of the trusted application;

a second portion including data indicating whether a particular one or more binaries can be loaded into the process space for the trusted application; and

a third portion derived from the data in both the first portion and the second portion by generating a digital signature over the first and second portions.

2. A computer readable medium as recited in claim 1, wherein the data structure, when populated with data, is a manifest corresponding to the trusted application, and wherein the unique identifier of the trusted application comprises:

a public key of a public-private key pair of a party that generates the manifest;

an identifier of the party that generates the manifest; and

a version number of the manifest.

3. A computer readable medium as recited in claim 1, wherein the data in the second portion comprises:

a list of one or more hashes of certificates that certify public keys which correspond to private keys that were used to sign the certificates that correspond to binaries that are authorized to execute in the process space.

4. A computer readable medium as recited in claim 3, wherein the data in the second portion further comprises:

a list of one or more additional hashes of certificates that certify public keys which correspond to private keys that were used to sign the certificates that correspond to binaries that are not authorized to execute in the process space.

5. A computer readable medium as recited in claim 1, wherein the data in the second portion comprises:

a list of one or more certificates that certify public keys which correspond to private keys that were used to sign the certificates that correspond to binaries that are authorized to execute in the process space.

6. A computer readable medium as recited in claim 5, wherein the data in the second portion further comprises:

a list of one or more additional certificates that certify public keys which correspond to private keys that were used to sign the certificates that correspond to binaries that are not authorized to execute in the process space.

7. A computer readable medium as recited in claim 1, wherein the data in the second portion comprises:

a list of one or more public keys which correspond to private keys that were used to sign the certificates that correspond to binaries that are authorized to execute in the process space.

8. A computer readable medium as recited in claim 7, wherein the data in the second portion further comprises:

a list of one or more public keys which correspond to private keys that were used to sign the certificates that correspond to binaries that are not authorized to execute in the process space.

9. A computer readable medium as recited in claim 1, wherein the data structure further comprises:

another portion that includes data representing a list of one or more export statements that allow a secret associated with the trusted application to be exported to another trusted application.

10. A computer readable medium as recited in claim 9, wherein the data structure, when populated with data, is a manifest corresponding to the trusted application, and wherein each of the one or more export statements comprises:

an identifier of the manifest;

an identifier of another manifest that corresponds to the trusted application to which the secret is to be exported; and

a digital signature over both the identifier of the manifest and the identifier of the other manifest.

11. A computer readable medium as recited in claim 10, wherein at least one of the one or more export statements comprises:

an identification of a particular computing device on which the at least one export statement is useable.

12. A computer readable medium as recited in claim 1, wherein the data structure further comprises:

another portion that includes data representing a set of properties corresponding to the data structure.

13. A computer readable medium as recited in claim 12, wherein the set of properties includes:

whether the trusted application is debuggable.

14. A computer readable medium as recited in claim 12, wherein the set of properties includes:

whether to allow an additional binary to be added to the process space after the trusted application begins executing.

15. A computer readable medium as recited in claim 12, wherein the set of properties includes:

whether to allow implicit upgrades to a higher version number.

16. A computer readable medium as recited in claim 1, wherein the data structure further comprises:

another portion that includes data representing a list of entry points into the executing trusted application.

17. A method of generating a new manifest to facilitate upgrading a trusted application on a computing device to a new trusted application, the method comprising:

receiving a request to upgrade the trusted application to the new trusted application;

receiving one or more new components to be included in the new trusted application; and

generating a manifest for the new trusted application, wherein the manifest allows the one or more new components to be loaded on the computing device.

18. A method as recited in claim 17, further comprising:

digitally signing each of the one or more new components.

19. A method as recited in claim 17, further comprising:

making the manifest available to a trusted core of an operating system executing on the computing device.

20. A method as recited in claim 17, wherein the manifest further prevents one or more components of the trusted application from being loaded on the computing device.

21. A method as recited in claim 17, wherein the manifest comprises:

a portion including data indicating whether a particular one or more binaries can be loaded into a process space for the new trusted application.

22. A method as recited in claim 17, wherein the manifest comprises:

a first portion including data representing a unique identifier of the new trusted application;

a second portion including data indicating whether a particular one or more binaries can be loaded into a process space for the new trusted application;

a third portion derived from the data in both the first portion and the second portion by generating a digital signature over the first and second portions;

a fourth portion that includes data representing a list of one or more export statements that allow a secret associated with the new trusted application to be exported to another trusted application; and

a fifth portion that includes data representing a set of properties corresponding to the manifest.

23. A method as recited in claim 17, wherein the manifest includes an identifier of the manifest, and wherein the identifier of the manifest includes:

a public key of a public-private key pair of a party that generates the manifest;

an identifier of the party that generates the manifest; and

a version number of the manifest.

24. A method as recited in claim 23, wherein:

an original manifest corresponds to the trusted application;

the public key of the manifest is the same as a public key in an identifier portion of the original manifest; and

the identifier of the party of the manifest is the same as an identifier, in the original manifest, of the party that generated the original manifest.

25. A method as recited in claim 17, wherein generating the manifest comprises:

adding, to the manifest, a list of one or more hashes of certificates that certify public keys which correspond to private keys that were used to sign the certificates that correspond to the one or more new components.

26. A method as recited in claim 17, wherein generating the manifest comprises:

adding, to the manifest, a list of one or more certificates that certify public keys which correspond to private keys that were used to sign the certificates that correspond to the one or more new components.

27. A method as recited in claim 17, further comprising:

adding, to the manifest, an indication of each of one or more additional components that cannot be executed in the process space.

28. A method as recited in claim 27, wherein adding, to the manifest, an indication of each of one or more additional components that cannot be executed in the process space, comprises:

adding, to the manifest, a list of one or more additional hashes of certificates that certify public keys which correspond to private keys that were used to sign the certificates that correspond to one or more additional components that are not authorized to execute in the process space.

29. A method as recited in claim 27, wherein adding, to the manifest, an indication of each of one or more additional components that cannot be executed in the process space, comprises:

adding, to the manifest, a list of one or more additional certificates that certify public keys which correspond to private keys that were used to sign the

certificates that correspond to one or more additional components that are not authorized to execute in the process space.

30. A method comprising:

receiving a request to execute a process;

setting up a memory space for the process;

accessing a manifest corresponding to the process; and

limiting which of a plurality of binaries can be executed in the memory space based on indicators, of the binaries, that are included in the manifest.

31. A method as recited in claim 30, wherein the manifest includes both a list of a first set of indicators of binaries that can be executed in the memory space and a list of a second set of indicators of binaries that cannot be executed in the memory space.

32. A method as recited in claim 30, further comprising receiving, with the request, the manifest.

33. A method as recited in claim 30, wherein limiting which of a plurality of binaries can be executed in the memory space comprises:

loading a set of binaries into the memory space;

checking whether each binary in the set is consistent with the manifest; and

not allowing any of the binaries in the set to be executed unless all binaries in the set are consistent with the manifest.

34. A method as recited in claim 33, wherein checking whether each binary in the set is consistent with the manifest comprises checking, for each binary, whether a certificate or a certificate hash corresponding to the binary is included in a list of authorized binaries and is not included in a list of non-authorized binaries.

35. A method as recited in claim 33, wherein limiting which of a plurality of binaries can be executed in the memory space further comprises:

allowing the binaries in the set to be executed if all binaries in the set are consistent with the manifest;

receiving a request to load an additional binary into the memory space;

checking whether the additional binary is consistent with the manifest; and

allowing the additional binary to be loaded into the memory space and executed if it is consistent with the manifest, otherwise not loading the additional binary into the memory space.

36. A method as recited in claim 30, wherein limiting which of a plurality of binaries can be executed in the memory space comprises, for each of the plurality of binaries:

checking whether the binary is consistent with the manifest;

loading the binary into the memory space if the binary is consistent with the manifest; and

not loading the binary into the memory space if the binary is inconsistent with the manifest.

37. A method as recited in claim 36, wherein checking whether the binary is consistent with the manifest comprises checking whether a certificate or a certificate hash corresponding to the binary is included in a list of authorized binaries and is not included in a list of non-authorized binaries.

38. A method as recited in claim 36, wherein limiting which of a plurality of binaries can be executed in the memory space further comprises:

receiving a request to load an additional binary into the memory space;

checking whether the additional binary is consistent with the manifest; and

allowing the additional binary to be loaded into the memory space and executed if it is consistent with the manifest, otherwise not loading the additional binary into the memory space.

39. A method as recited in claim 30, wherein memory space comprises a virtual memory space.

40. A method as recited in claim 30, wherein the manifest comprises data indicating whether a particular one or more binaries can be loaded into the memory space for the process.

41. A method as recited in claim 30, wherein the manifest comprises:

a first portion including data representing a unique identifier of the process;

a second portion including data indicating whether a particular one or more binaries can be loaded into the memory space for the process;

a third portion derived from the data in both the first portion and the second portion by generating a digital signature over the first and second portions;

a fourth portion that includes data representing a list of one or more export statements that allow a secret associated with the process to be exported to another process; and

a fifth portion that includes data representing a set of properties corresponding to the manifest.

42. A method as recited in claim 30, further comprising:

receiving, from the process, a request to securely store a secret, wherein the request includes,

the secret,

a public key of a public-private key pair of a party that generated a manifest for the process,

an identifier of the party, and

a set of one or more versions of the manifest that should be allowed to retrieve the secret; and

having the secret encrypted.

43. A method as recited in claim 30, further comprising:

receiving, from the process, a request to securely store a secret, wherein the request includes,

the secret, and

an identifier of one or more manifests that should be allowed to retrieve the secret; and

having the secret encrypted.

44. A method as recited in claim 30, further comprising:

receiving, from the process, a request to retrieve a secret securely stored by a previous process;

comparing the manifest identifier of the requesting process to a collection of one or more manifest identifiers with which the secret was originally sealed; and

determining whether to reveal the secret to the process based at least in part on whether the manifest identifier of the requesting process and one of the collection of manifest identifiers are the same.

45. A method as recited in claim 30, further comprising:

receiving encrypted data;

decrypting the data;

identifying a plurality of conditions in the data;

checking whether the manifest satisfies all of the plurality of conditions; and

allowing the process to retrieve a secret in the encrypted data only if the manifest satisfies all of the plurality of conditions.

46. A method as recited in claim 30, further comprising:

receiving encrypted data;

decrypting the data;

identifying one or more conditions in the data;

checking whether the data satisfies the one or more conditions; and

allowing the process to retrieve a secret in the encrypted data only if the data satisfies the one or more conditions.

47. A method as recited in claim 30, further comprising:

receiving, from the process, a request to generate a digitally signed statement; and

generating a digitally signed statement including an identifier of the manifest corresponding to the process.

48. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to:

set up a virtual memory space for a trusted application process;

obtain a manifest corresponding to the trusted application process;

identify, from the manifest, a plurality of binary indicators; and

restrict which of multiple binaries can be executed in the virtual memory space based on the plurality of binary indictors.

49. One or more computer readable media as recited in claim 48, wherein the instructions that cause the one or more processors to restrict which of multiple binaries can be executed in the virtual memory space cause the one or more processors to:

load a set of binaries into the virtual memory space;

check whether each binary in the set is consistent with the manifest; and

not allow any of the binaries in the set to be executed unless all binaries in the set are consistent with the manifest.

50. One or more computer readable media as recited in claim 49, wherein the instructions that cause the one or more processors to check whether each binary in the set is consistent with the manifest cause the one or more processors to check, for each binary, whether a certificate or a certificate hash corresponding to the binary is included in a list of authorized binaries and is not included in a list of non-authorized binaries.

51. One or more computer readable media as recited in claim 49, wherein the instructions that cause the one or more processors to restrict which of multiple binaries can be executed in the virtual memory space cause the one or more processors to:

allow the binaries in the set to be executed if all binaries in the set are consistent with the manifest;

receive a request to load an additional binary into the virtual memory space;

check whether the additional binary is consistent with the manifest; and

allow the additional binary to be loaded into the virtual memory space and executed if it is consistent with the manifest, and otherwise not load the additional binary into the virtual memory space.

52. One or more computer readable media as recited in claim 48, wherein the instructions that cause the one or more processors to restrict which of multiple binaries can be executed in the virtual memory space cause the one or more processors to, for each of the multiple binaries:

check whether the binary is consistent with the manifest;

load the binary into the virtual memory space if the binary is consistent with the manifest; and

not load the binary into the virtual memory space if the binary is inconsistent with the manifest.

53. One or more computer readable media as recited in claim 52, wherein the instructions that cause the one or more processors to check whether the binary is consistent with the manifest cause the one or more processors to check whether a certificate or a certificate hash corresponding to the binary is included in a list of authorized binaries and is not included in a list of non-authorized binaries.

54. One or more computer readable media as recited in claim 48, wherein the manifest comprises data indicating whether a particular one or more binaries can be loaded into the virtual memory space for the trusted application process.

55. One or more computer readable media as recited in claim 48, wherein the manifest comprises:

a first portion including data representing a unique identifier of the trusted application process;

a second portion including data indicating whether a particular one or more binaries can be loaded into the virtual memory space for the trusted application process;

a third portion derived from the data in both the first portion and the second portion by generating a digital signature over the first and second portions;

a fourth portion that includes data representing a list of one or more export statements that allow a secret associated with the trusted application process to be exported to another trusted application process; and

a fifth portion that includes data representing a set of properties corresponding to the manifest.

56. One or more computer readable media having stored thereon a plurality of instructions to implement a trusted core of a computing device that, when executed by one or more processors of the computing device, causes the one or more processors to:

receive, from a trusted agent executing on the computing device, a request to securely store a secret, wherein the request includes,

the secret, and

an identifier of a manifest that should be allowed to retrieve the secret; and

have the secret encrypted.

57. One or more computer readable media as recited in claim 56, wherein the identifier comprises:

a public key of a public-private key pair of a party that generated the manifest for the trusted agent;

an identifier of the party; and

a set of one or more versions of the manifest that should be allowed to retrieve the secret.

58. One or more computer readable media having stored thereon a plurality of instructions to implement a trusted core of a computing device that, when executed by one or more processors of the computing device, causes the one or more processors to:

receive, from a trusted application executing on the computing device, a request to retrieve a secret securely stored by a previous trusted application executing on the computing device;

compare a first manifest identifier of the trusted application to a second manifest identifier corresponding to the previous trusted application; and

determine whether to reveal the secret to the trusted application based at least in part on whether the first manifest identifier and the second manifest identifier are the same.

59. One or more computer readable media as recited in claim 58, wherein the trusted application is an upgraded version of the previous trusted application.

60. One or more computer readable media as recited in claim 58, wherein:

the first manifest includes a first public key of a first public-private key pair of a party that generated the first manifest, an identifier of the party that generated the first manifest, and a version indicator of the first manifest; and

the second manifest includes a second public key of a second public-private key pair of a party that generated the second manifest, an identifier of the party that generated the second manifest, and a version indicator of the second manifest.

61. One or more computer readable media as recited in claim 60, wherein the instructions that cause the one or more processors to determine whether to reveal the secret to the trusted application cause the one or more processors to:

check whether the first public key is the same as the second public key and whether the identity of the party that generated the first manifest is the same as the identity of the party that generated the second manifest;

check whether the version indicator of the first manifest is the same as one or more version indicators supplied by the previous trusted application when securely storing the secret; and

refuse to reveal the secret to the trusted application if the checking indicates any one or more of the following: the first public key is not the same as the second public key, the identity of the party that generated the first manifest is not the same as the identity of the party that generated the second manifest, the version indicator of the first manifest is not the same as one or more version indicators supplied by the previous trusted application when securely storing the secret.

62. One or more computer readable media as recited in claim 58, wherein the instructions that cause the one or more processors to determine whether to reveal the secret to the trusted application further cause the one or more processors to reveal the secret to the trusted application based at least in part on whether an export certificate corresponding to the previous trusted application identifies the first manifest as being authorized to retrieve the secret.

63. One or more computer readable media as recited in claim 62, wherein the export certificate includes:

an identification of the first manifest;

an identification of the second manifest, wherein the second manifest was digitally signed using a private key of a public-private key pair of a party that generated the second manifest; and

a digital signature over the identification of the first manifest and the identification of the second manifest, wherein the digital signature is generated using the private key.

64. One or more computer readable media having stored thereon a plurality of instructions to implement a trusted core of a computing device that, when executed by one or more processors of the computing device, causes the one or more processors to:

receive encrypted data;

decrypt the data;

identify a plurality of conditions in the data;

check whether a manifest associated with a trusted application process satisfies all of the plurality of conditions; and

allow the trusted application process to retrieve a secret in the encrypted data only if the manifest satisfies all of the plurality of conditions.

65. One or more computer readable media as recited in claim 64, wherein:

the plurality of conditions include a public key of a public-private key pair of a party that needs to have digitally signed the manifest; and

the instructions that cause the one or more processors to check whether the manifest associated with the trusted application process satisfies all of the plurality of conditions cause the one or more processors to check whether the manifest includes the public key.

66. One or more computer readable media as recited in claim 64, wherein:

the plurality of conditions include an identifier of a party that needs to have generated the manifest; and

the instructions that cause the one or more processors to check whether the manifest associated with the trusted application process satisfies all of the plurality of conditions cause the one or more processors to check whether the manifest was generated by the party.

67. One or more computer readable media as recited in claim 64, wherein:

the plurality of conditions include one or more versions that the manifest needs to be; and

the instructions that cause the one or more processors to check whether the manifest associated with the trusted application process satisfies all of the plurality of conditions cause the one or more processors to check whether the manifest is one of the one or more versions.

68. One or more computer readable media as recited in claim 64, wherein the plurality of instructions further cause the one or more processors to:

check whether the data satisfies one or more of the plurality of conditions; and

allow the trusted application process to retrieve the secret only if the data satisfies the one or more conditions.

69. One or more computer readable media having stored thereon a plurality of instructions to implement a trusted core of a computing device that, when executed by one or more processors of the computing device, causes the one or more processors to:

receive, from a trusted application, a request to generate a digitally signed statement; and

generate a digitally signed statement including an identifier of a manifest corresponding to the trusted application.

70. One or more computer readable media as recited in claim 69, wherein the plurality of conditions include a public key of a public-private key pair of a party that digitally signed the manifest.

71. One or more computer readable media as recited in claim 69, wherein the plurality of conditions include an identifier of a party that generated the manifest.

72. One or more computer readable media as recited in claim 69, wherein the plurality of conditions include one or more manifest version indicators.

73. A computer readable medium having stored thereon a data structure that allows a secret associated with a trusted application to be exported to another trusted application, the data structure comprising:

a first portion including an identifier of a manifest associated with the application;

a second portion including an identifier of a manifest associated with the other application; and

a third portion derived from the identifiers in both the first portion and the second portion by generating a digital signature over the first and second portions.

74. A computer readable medium as recited in claim 73, wherein the data structure further comprises:

a fourth portion including an identification of a particular computing device on which the data structure can be used to export the secret.

75. A computer readable medium as recited in claim 73, wherein the second portion includes identifiers of a plurality of manifests associated with a plurality of additional applications to which the secret can be exported.

76. A computer readable medium as recited in claim 75, wherein each of the plurality of additional applications is a newer version of the other application.